



# CMS32F033 应用笔记

## I2C 应用笔记

Rev. 1.0.2

请注意以下有关CMS知识产权政策

\* 中微半导体（深圳）股份有限公司（以下简称本公司）已申请了专利，享有绝对的合法权益。与本公司MCU或其他产品有关的专利权并未被同意授权使用，任何经由不当手段侵害本公司专利权的公司、组织或个人，本公司将采取一切可能的法律行动，遏止侵权者不当的侵权行为，并追讨本公司因侵权行为所受的损失、或侵权者所得的不法利益。

\* 中微半导体（深圳）股份有限公司的名称和标识都是本公司的注册商标。

\* 本公司保留对规格书中产品在可靠性、功能和设计方面的改进作进一步说明的权利。然而本公司对于规格内容的使用不负责任。文中提到的应用其目的仅仅是用来做说明，本公司不保证和不表示这些应用没有更深入的修改就能适用，也不推荐它的产品使用在会由于故障或其它原因可能会对人身造成危害的地方。本公司的产品不授权适用于救生、维生器件或系统中作为关键器件。本公司拥有不事先通知而修改产品的权利，对于最新的信息，请参考官方网站 [www.mcu.com.cn](http://www.mcu.com.cn)

## 目录

<b>1</b>	<b>概述</b>	<b>3</b>
<b>2</b>	<b>功能特性</b>	<b>3</b>
<b>3</b>	<b>功能描述</b>	<b>4</b>
3.1	数据有效性	4
3.2	START 与 Repeated START	4
3.3	STOP	5
3.4	数据传输格式	5
3.5	ACK 与 NACK	5
3.6	时钟	6
3.7	时钟同步	6
3.8	主机仲裁	7
3.9	从机地址	7
3.9.1	7 位地址模式	8
3.9.2	10 位地址模式	9
3.9.3	地址 Mask 功能	9
<b>4</b>	<b>工作模式</b>	<b>10</b>
4.1	I2C 模块中断与状态码	10
4.2	主机模式	10
4.3	从机模式	13
4.4	广播地址模式	14
4.5	唤醒模式	14
<b>5</b>	<b>总结</b>	<b>15</b>
5.1	I2C 从机地址	15
5.1.1	7 位地址模式	15
5.1.2	10 位地址模式	15
5.2	I2C 时钟设置	15
5.3	I2C 总线错误状态	16
5.3.1	总线错误状态码	16
5.3.2	仲裁与总线错误	16
5.4	总线清除 (Bus clear)	16
<b>6</b>	<b>版本修订说明</b>	<b>17</b>

## 1 概述

I2C 是一种两线双向串行总线，为设备之间的数据交换提供了一种简单有效的连接方式（如图 1-1）。

I2C 是一个真正的多主机总线，包含了冲突检测和仲裁机制。冲突检测和仲裁机制用来在两个或多个主机同时尝试控制总线的情况下，防止数据损坏。

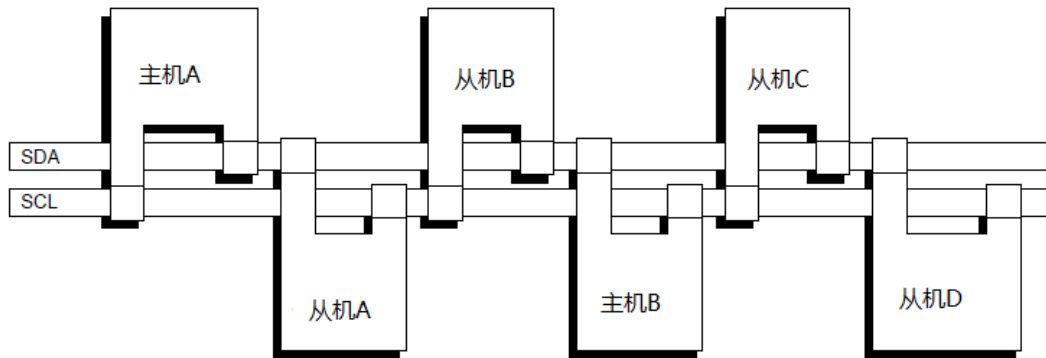


图 1-1: I2C 总线连接示意图

## 2 功能特性

- ◆ 支持主机/从机模式。
- ◆ 主从机之间双向数据传送。
- ◆ 多主机总线。
- ◆ 多主机间同时传输数据仲裁，避免总线上串行数据损坏。
- ◆ 总线采用串行同步时钟，可实现设备之间以不同的速率传输。
- ◆ 串行同步时钟可以被用来作为握手机制实现挂起和恢复串行传输。
- ◆ 可编程的时钟可以用于多种速率控制。
- ◆ 支持 7 位/10 位从地址模式。
- ◆ 支持多地址识别（4 组从机地址带 mask 选项）。
- ◆ 支持唤醒模式。

## 3 功能描述

### 3.1 数据有效性

SDA 线上的数据在 SCL 的高电平时间内必须保持稳定，SDA 的电平状态可在 SCL 的低电平时间内改变。每传输一位数据产生一个时钟。

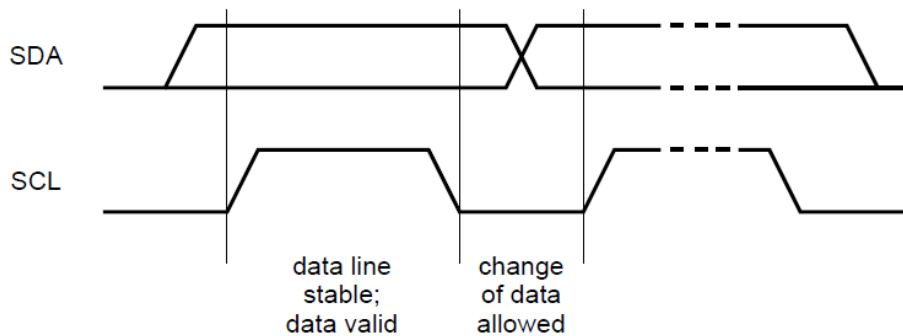


图 3-1: I2C 数据有效性

### 3.2 START 与 Repeated START

I2C 总线上数据的传输以 START（启动）信号开始，I2C 总线便进入忙碌状态。主机可以发送 Repeated START（重复启动信号）改变数据的发送方向（如：写入从机改变为读取从机），并且 I2C 总线继续保持忙碌状态。

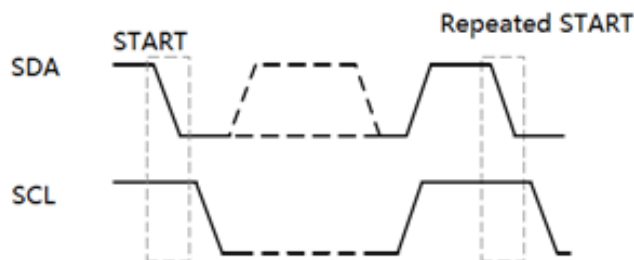


图 3-2: START 与 Repeated START

如图 3-2 所示，当 I2C 总线空闲时，SDA 与 SCL 在上拉电阻的作用下呈现出高电平。此时主机可发送 START 信号开始新的数据传输。

当 SCL 为高电平时，SDA 从高电平跳变到低电平代表 START（启动）信号。Repeated START（重复启动）信号与 START 信号产生条件相同。

### 3.3 STOP

当 SCL 为高电平时，SDA 从低电平跳变到高电平代表 STOP（停止）信号。意味数据传输的结束，I2C 总线进入空闲状态。

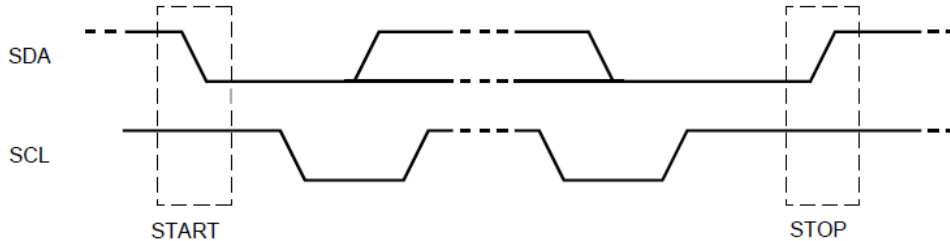


图 3-3: START 与 STOP

### 3.4 数据传输格式

I2C 总线上以 8 位长度的数据（1Byte）作为传输单位，每个 Byte 后紧随着 1 位应答位。数据以高位字节（MSB）在前，低位字节（LSB）在后的顺序进行传输。

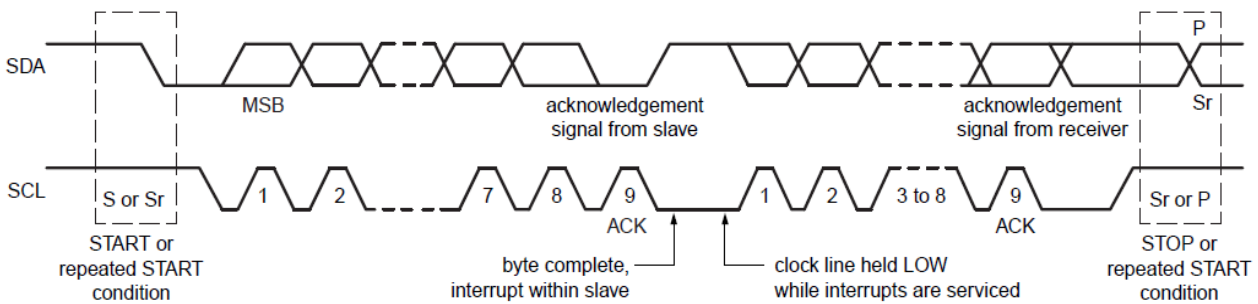


图 3-4: 数据传输（转载自网络）

### 3.5 ACK 与 NACK

每个 Byte 数据发送后都会紧跟着第 9 个 SCL 时钟信号传输 ACK（应答）信号。在第 9 个 SCL 高电平区间，SDA 为低电平代表有应答信号，如果 SDA 为高电平，代表无应答信号。

如图 3-4 所示，在主机寻址或者传输数据到从机时，主机可根据从机是否回复 ACK 判断从机的状态。如果从机未回复 ACK 信号，主机可以发送 STOP 信号结束传输或者发送 Repeated START 信号开启新的数据传输。在主机接收数据时，不发送 ACK 信号给从机，从机将会释放 SDA 控制权，主机可发送 STOP 信号结束传输和 Repeated START 信号开启新的数据传输。

### 3.6 时钟

SCL 时钟频率的可通过设置 I2C 时钟控制寄存器 I2CCLK 设置。

I2C 采样时钟 = Pclk / 2<sup>M</sup> (M = 0~7)

SCL 时钟 = Pclk / (2<sup>M</sup> \* (N+1) \* 10) (N = 0~15)

详细的寄存器说明可参考《CMS32F033 用户手册》

表 3-1: I2C 时钟配置参考

I2C 模式	Pclk (APB 时钟频率)	M	N
100K(标准模式)	48MHz	2	11
200K	48MHz	1	11
400K(快速模式)	48MHz	0	11

### 3.7 时钟同步

两个主机可以同时在一个 I2C 总线上传输，通过时钟同步和仲裁决定哪一个主机控制总线并完成其传输。在单个主机的系统中时不需要同步和仲裁。

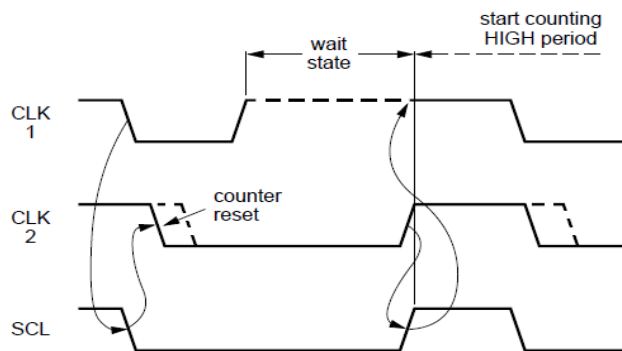


图 3-5: 时钟同步 (转载自网络)

如图 3-5 所示，SCL 的电平是主机 1 与主机 2 的时钟信号相与的结果。

### 3.8 主机仲裁

仲裁功能保证了数据正确传输。

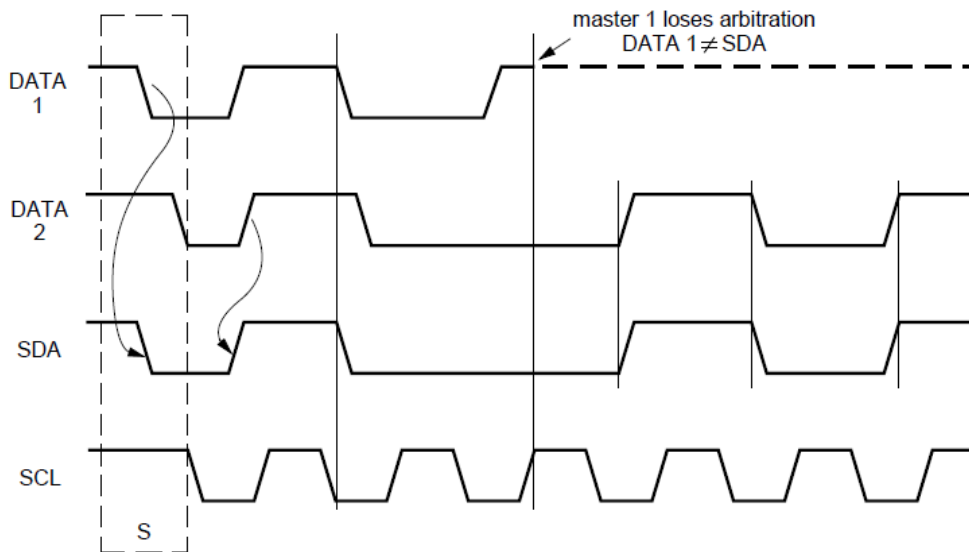


图 3-6：仲裁（转载自网络）

如图 3-6 所示，SDA 的电平是主机 1 和主机 2 的 DATA 相与的结果。主机每次发送数据时都会检测数据是否正确发送。当主机检测到发送的数据与 SDA 上的数据不匹配时，主机就会丢失仲裁权。如图 3-6 中主机 1 发送的数据与 SDA 不匹配，主机 1 丢失仲裁权。当主机 1 在发送数据时丢失仲裁权，主机 1 在丢失仲裁后完成剩余时钟的发送后便进入从机模式。赢得仲裁权的主机 2 将会继续发送数据。

当主机发生仲裁失败后，软件代码上需要初始化整个数据帧。若仲裁失败并匹配了设备地址，则该主机设备将变为从机设备。

### 3.9 从机地址

I2C 模块支持 7 位地址模式、10 位地址模式。I2C 模块有 4 个 7 位地址寄存器 and 对应的地址 Mask 寄存器、1 个 10 位地址寄存器和对应的地址 Mask 寄存器。可支持 4 个 7 位从机地址与 1 个 10 位从机地址。通过地址 Mask 功能，可以根据需求组合成不同的从机地址。

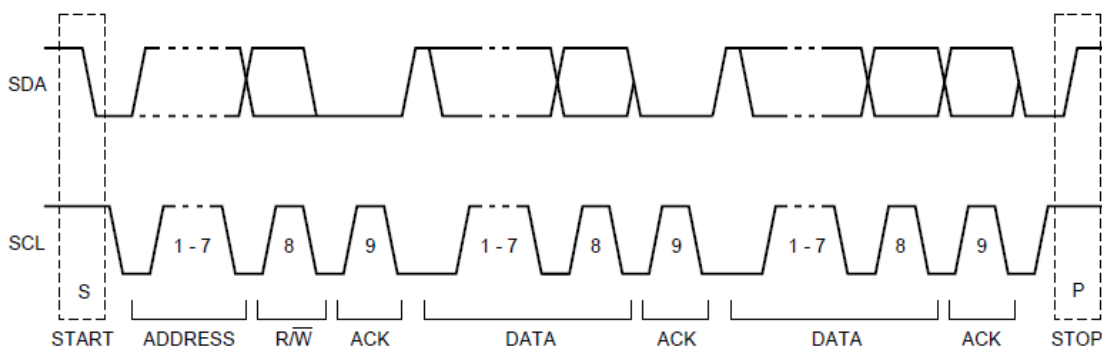


图 3-7：数据传输（转载自网络）

### 3.9.1 7 位地址模式

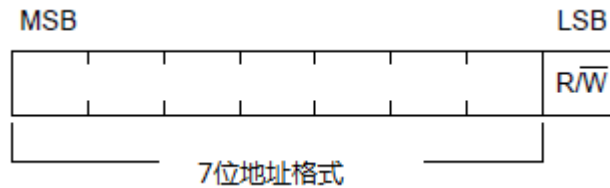


图 3-8: 7 位地址格式

图 3-9 展示了主机发送数据到从机的过程。

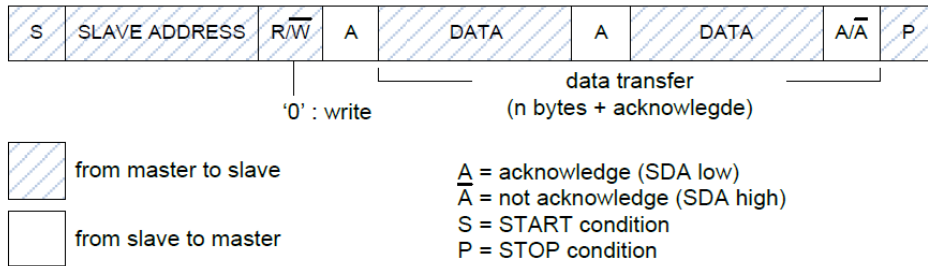


图 3-9: 主机发送数据 (转载自网络)

图 3-10 展示了主机从从机读取数据的过程。

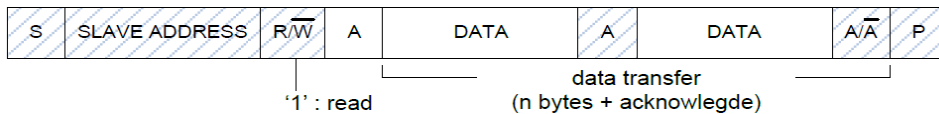


图 3-10: 主机读取数据 (转载自网络)

图 3-11 展示了一个主机与从机的通讯过程。

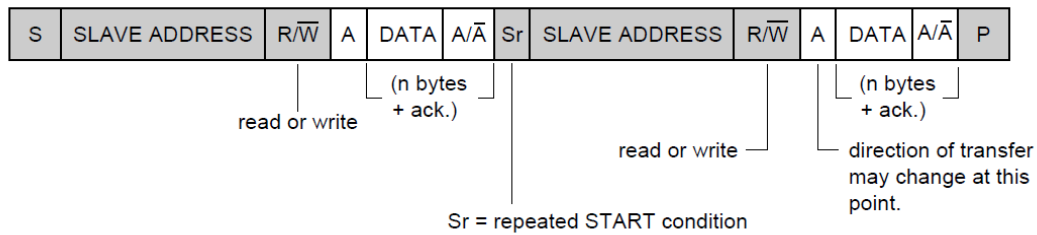


图 3-11: 主机与从机的通讯过程 (转载自网络)



### 3.9.2 10 位地址模式

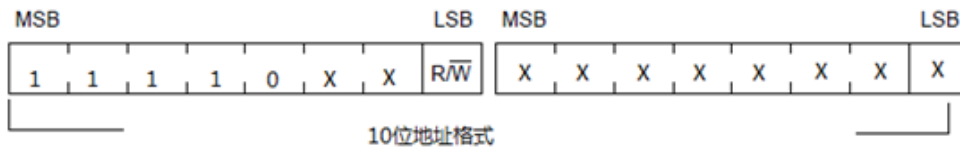


图 3-12: 10 位地址格式 (转载自网络)

图 3-13 展示了主机发送数据到从机的通迅过程。

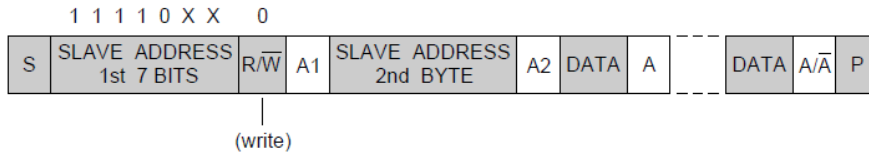


图 3-13: 主机发送数据 (转载自网络)

图 3-14 展示了主机从从机读取数据的过程。

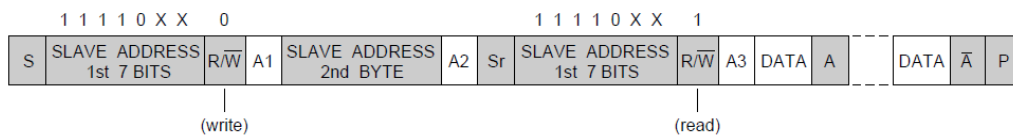


图 3-14: 主机读取数据 (转载自网络)

图 3-15 展示了主机与从机的通迅过程

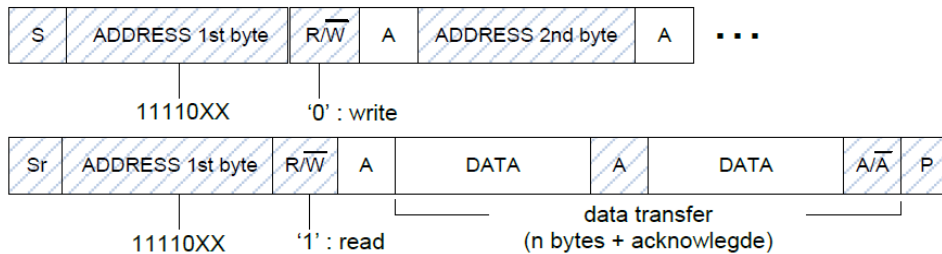


图 3-15: 主机与从机的通迅过程 (转载自网络)

### 3.9.3 地址 Mask 功能

I2C 模块支持地址 Mask 功能。根据从机地址掩码寄存器 (I2CADMn) 的设置, 在从机模式下匹配地址时可选择 不匹配该地址位。如从机地址为 B'1110110', 地址掩码寄存器的值为 B'0000110' (0: 不比较该位地址; 1: 比较该位地址), 则从机的地址变为 B'XXXX110', 则只要主机发送的从机地址的低 3 位为 '110', 该从机就会与该地址匹配。详细信息可参考《CMS32F033 用户手册》。

## 4 工作模式

I2C 模块可根据 I2C 总线上的状态切换主机/从机模式。

### 4.1 I2C 模块中断与状态码

I2C 模块一共有 30 种状态位。其中包括 29 种与中断相关的状态、1 种与中断无关的状态 0xF8。详细信息可参考《CMS32F033 用户手册》。产生中断时，SI 位置位，可通过置位 SIC 位清除 SI 位。

### 4.2 主机模式

通过 I2C 时钟控制寄存器(I2CCLK)配置 I2C 模块的 SCL 时钟,设置 I2C 控制位寄存器(I2CCONSET[Bit6])的 I2CEN 位使能 I2C 接口,同时设置 IO 配置寄存器(IOPxxCONFIG)口复用为 I2C 数字功能。

通过置位 STA 位 (I2CCONSET[Bit5])使 I2C 模块进入主机模式并发送 START 信号。当 START 信号发送完成时, STA 位自动清 0。

置位 AAC (I2CCONCLR[Bit2]), 主机不会应答。在读取从机数据时, 从机监测到主机没有应答便会释放 SDA 控制权。当主机仲裁失败时, 主机进入从机模式, 若匹配到 I2C 总线上的地址或者广播地址, 也不会应答。

主机可通过发送从机的地址实现与从机的通讯, 数据发送的过程可参考 3.9 章节。

表 4-1: 状态码

I2C 总线与 I2C 硬件状态	产生的状态码	软件代码操作					I2C 硬件动作
		I2CDAT	SIC	AAC/AA	STA	STO	
发送 START	0x08	写入从机地址+写	1	AA=0 AAC=0	0	0	主机发送 I2CDAT 寄存器的值 (从机地址+写位) 并接收从机 ACK
		写入从机地址+读	1	AA=0 AAC=0	0	0	主机发送 I2CDAT 寄存器的值 (从机地址+读位), 接收从机 ACK 并进入接收模式
发送 Repeated STAT (重启信号)	0x10	写入从机地址+写	1	AA=0 AAC=0	0	0	主机发送 I2CDAT 寄存器的值 (从机地址+写位) 并接收从机 ACK
		写入从机地址+读	1	AA=0 AAC=0	0	0	主机发送 I2CDAT 寄存器的值 (从机地址+读位), 接收从机 ACK 并进入接收模式
发送地址+写完成并接收到从机 ACK 或者发送地址+写完成未接收到从机 ACK	0x18 或者 0x20	写入数据	1	AA=0 AAC=0	0	0	主机发送 I2CDAT 寄存器的值并接收从机 ACK
		X	1	AA=0 AAC=0	1	0	主机发送 Repeated STAT
		X	1	AA=0 AAC=0	0	1	主机发送停止信号, 当停止信号发送完成时, STO 位

I2C 总线与 I2C	产生的状态码	软件代码操作					自动清 0
		X	1	AA=0 AAC=0	1	1	主机先发送一个STOP信号，接着发送一个START信号
发送数据完成并接收到从机 ACK 或者发送数据完成未接收到从机 ACK	0x28 或者 0x30	写入数据	1	AA=0 AAC=0	0	0	主机发送 I2CDAT 寄存器的值并接收从机 ACK
		X	1	AA=0 AAC=0	1	0	主机发送 Repeated STAT
		X	1	AA=0 AAC=0	0	1	主机发送停止信号，当停止信号发送完成时，STO 位自动清 0
		X	1	AA=0 AAC=0	1	1	主机先发送一个STOP信号，接着发送一个START信号
发送地址+读完成并接收到从机 ACK 或者发送地址+读完成未接收到从机 ACK	0x40 或者 0x48	X	1	AA=1 AAC=0	0	0	主机接收数据并回复 ACK
		X	1	AA=0 AAC=1	0	0	主机接收数据，不回复 ACK
		X	1	AA=0 AAC=0	1	0	主机发送 Repeated STAT
		X	1	AA=0 AAC=0	0	1	主机发送停止信号，当停止信号发送完成时，STO 位自动清 0
		X	1	AA=0 AAC=0	1	1	主机先发送一个 STOP 信号，接着发送一个 START 信号
接收数据完成并接收到从机 ACK 或者接收数据完成未接收到从机 ACK	0x50 或者 0x58	X	1	AA=1 AAC=0	0	0	主机接收数据并回复 ACK
		X	1	AA=0 AAC=1	0	0	主机接收数据，不回复 ACK
		X	1	AA=0 AAC=0	1	0	主机发送 Repeated STAT
		X	1	AA=0 AAC=0	0	1	主机发送停止信号，当停止信号发送完成时，STO 位自动清 0
		X	1	AA=0 AAC=0	1	1	主机先发送一个 STOP 信号，接着发送一个 START 信号
地址或数据传输过程中仲裁失败	0x38	X	1	AA=1 AAC=0	0	0	进入从机模式，若被寻址，则应答。
		X	1	AA=0 AAC=1	0	0	进入从机模式，若被寻址，则不应答，主机进入不可寻址的从机模式

I2C 总线与 I2C	产生的状态码	软件代码操作					
主机仲裁失败, 接收到从机地址+写并回复 ACK	0x68	X	1	X	X	X	从此主机将作为一个从设备。详细可参考从机模式。
主机仲裁失败, 接收到从机地址+读并回复 ACK	0xB0	X	1	X	X	X	
主机仲裁失败, 接收到广播呼叫地址 (主机开启广播模式), 回复 ACK	0x78	X	1	X	X	X	
发送完第二个地址, 接收到 ACK (10 位地址) 或者 发送完第二个地址, 未接收到 ACK (10 位地址)	0xE0 或者 0xE8	写入数据	1	AA=0 AAC=0	0	0	主机发送 I2CDAT 寄存器的值并接收从机 ACK
		X	1	AA=0 AAC=0	1	0	主机发送 Repeated STAT
		X	1	AA=0 AAC=0	0	1	主机发送停止信号, 当停止信号发送完成时, STO 位自动清 0
		X	1	AA=0 AAC=0	1	1	主机先发送一个 STOP 信号, 接着发送一个 START 信号
总线错误	0x00	X	1	X	X	1	主机重置

在主机模式下置位 STA 后, 主机发送 START 信号, 发送完成 (I2C 总线上出现 START 信号) 则 STA 位自动清零。若没有发送完成, 则可以置位 STAC 清除 STA, 使主机恢复正常。

在主机模式下置位 STO 后, 主机发送 STOP 信号, 发送完成 (I2C 总线上出现 STOP 信号) 则 STO 位自动清零。若没有发送完成, 则可以将 0x7 写入 I2C 软件复位寄存器 (I2CRST), 使主机恢复正常, 但 I2C 模块的所有寄存器将会复位。

### 4.3 从机模式

I2C 模块支持 4 个 7 位从机地址和 1 个 10 位地址，并且从机地址带 Mask 功能，每个地址都可选择是否开启广播地址模式。置位 I2CEN 位使能 I2C 接口，同时设置 IO 配置寄存器(IOPxxCONFIG)口复用为 I2C 数字功能。

置位 AA 位，开启从机应答。否则从机将会进入不可寻址模式。

表 4-2: 状态码

I2C 总线与 I2C 硬件状态	产生的状态码	软件代码操作					I2C 硬件动作
		I2CDAT	SIC	AAC /AA	STA	STO	
从机模式下接收到地址 +写位并回复了 ACK	0x60	x	1	AA=1 AAC=0	0	0	从机接收数据
		x	1	AA=0 AAC=1	0	0	从机接收数据，并不回复 ACK
从机地址匹配后接收到 数据，回复 ACK	0x80	读取数据	1	AA=1 AAC=0	0	0	从机接收数据，并回复 ACK
		读取数据	1	AA=0 AAC=1	0	0	从机接收数据，不回复 ACK
从机地址匹配后接收到 数据，不回复 ACK	0x88	读取数据	1	AA=0 AAC=1	0	0	从机不再响应主机，且不 可寻址
从机模式下接收到地址 +读位，回复 ACK	0xA8	写入数据	1	AA=1 AAC=0	0	0	从机发送数据
从机模式下发送数据 后，接收到 ACK	0xB8	写入数据	1	AA=1 AAC=0	0	0	从机发送数据
		写入数据	1	AA=0 AAC=1	0	0	从机发送最后一个数据
从机模式下发送数据 后，未接收到 ACK	0xC0	X	1	AA=1 AAC=0	0	0	主机不再读取从机数据， 从机不发送数据
从机模式下发送完最后 一个数据，接收到 ACK	0xC8	X	1	AA=1 AAC=0	0	0	从机不再发送数据
从机模式下发送完最后 一个数据，未接收到	0xD0	X	1	AA=1	0	0	从机不再发送数据

I2C 总线与 I2C 硬件状	产生的状态码	软件代码操作					
ACK				AA=0			
接收到广播呼叫地址， 回复 ACK	0x70	X	1	AA=1 AA=0	0	0	接收数据，并回复 ACK
		X	1	AA=0 AA=1	0	0	接收数据，不回复 ACK
从机接收广播呼叫地址 后接收到数据，回复 ACK	0x90	读取数据	1	AA=1 AA=0	0	0	接收数据，回复 ACK
		读取数据	1	AA=0 AA=1	0	0	接收数据，不回复 ACK
从机接收广播呼叫地址 后接收到数据，不回复 ACK	0x98	读取数据	1	AA=1 AA=0	0	0	主机发送数据或者寻址， 从机将会回复 ACK
		读取数据	1	AA=0 AA=1	0	0	主机发送数据或者寻址， 从机将不会回复 ACK
从机模式下接收到停止 信号或重新启动信号	0xA0	X	1	AA=1 AA=0	0	0	主机发送数据或者寻址， 从机将会回复 ACK
		X	1	AA=0 AA=1	0	0	主机发送数据或者寻址， 从机将不会回复 ACK
以上任一状态	X	X	1	X	1	0	结束当前传输并等待总线 空闲时进入主机模式，并 发送START信号
以上任一状态	X	X	1	X	0	1	I2C模块会当作接收到一 个停止位 并产生0xA0状态位和中断 标志。

## 4.4 广播地址模式

置位 I2C 模块从机地址寄存器的 GC 位可开启广播地址模式。当主机发送地址 0x00 时，开启广播地址模式的从机将会被寻址，并且进入接收模式。

## 4.5 唤醒模式

I2C 从机模式下，当 I2C 从机地址匹配时，芯片会从睡眠模式中唤醒。

I2C 唤醒模式不支持深度睡眠和停止模式。

## 5 总结

### 5.1 I2C 从机地址

#### 5.1.1 7 位地址模式

I2C 模块在从机模式下，需要设置 I2C 从机地址。I2C 模块共拥有 4 个 7 位从机地址寄存器和 1 个 10 位地址寄存器。

在使用 7 位地址时，需要同时配置 4 个 7 位从机地址寄存器。可将 4 个寄存器配置为 4 个不同的地址。当使用 1 个地址时，需要将 4 个 7 位寄存地址寄存器设置为相同的值。应为芯片复位会将地址寄存器的值设置为 0x00，若不设置地址寄存器会导致从机响应广播地址（0x00），尽管此时并没有开启广播地址模式。

#### 5.1.2 10 位地址模式

在进行 10 位地址的通讯时，主机先发送从机地址高位，产生状态码 0x18(地址+写位发送完成，接收到 ACK)或者 0x20（地址+写位发送完成，未接收到 ACK）。若状态码为 0x18，主机可发送从机低位地址，产生状态码 0xE0（主机模式下发送完第二个地址，接收到 ACK）或者 0xE8（主机模式下发送完第二个地址，未接收到 ACK）。

### 5.2 I2C 时钟设置

《CMS32F033 用户手册》提及的 I2C 时钟控制寄存器（I2CCLK）描述中：

- 采样时钟= PCLK/2^M。
- SCL 时钟= PCLK/（2^M ×（N+1）×10）。

## 5.3 I2C 总线错误状态

### 5.3.1 总线错误状态码

经过查阅相关资料以及其他公司芯片 I2C 模块资料，I2C 总线错误状态：在主机在发送数据过程中，出现了非法的 START 或者 STOP 信号（如图 5-1），I2C 主机将会产生总线错误状态码（0x00）与中断。

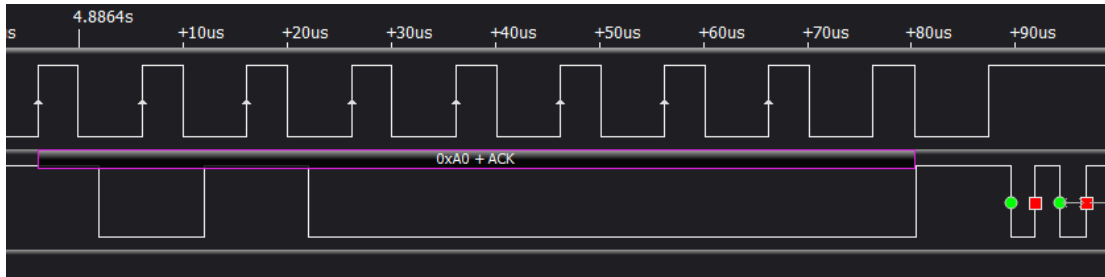


图 5-1：非法的 START 与 STOP 信号

### 5.3.2 仲裁与总线错误

在实际使用 I2C 主机模式时，I2C 总线上出现图 5-1 的传输波形时并非一定是 I2C 总线错误，也会产生 0xA0（从机模式下接收到停止信号或重启动信号）状态码。

可能的原因：

1) I2C 仲裁模式

I2C 主控模式下，I2C 主机发送数据时会检测 SDA 总线上的值是否与发送的数据一致。若不一致，则会产生仲裁。

2) I2C 主控模式下对 SDA 的采样

从《CMS32F033 用户手册》上可知，配置 I2C 模式时需要配置 采样时钟。假设此采样时钟指的是主机模式下对 SDA 的采样时钟。

3) I2C 仲裁失败

若 I2C 主机仲裁失败，则 I2C 主机进入从机模式，发送完剩余的时钟并产生中断和状态码。

如图 5-1 所示，若主机先检测到 SDA 数据并非发送的数据（SDA 的高/低电平），产生仲裁失败并进入从机模式。若此时总线上出现停止位或在重启位，则产生状态码 0xA0（从机模式下接收到停止信号或重启动信号）。

若主机先检测到非法的 STOP 或者 START 信号（SDA 的上升沿或者下降沿），则产生总线错误中断。

当然此结论是根据查阅的资料和主机的采样模式来推测的。若此推论成立，则增加采样频率，则总线错误状态出现的频率将会提高一些。

## 5.4 总线清除（Bus clear）

在外界干扰或者其他异常的事件导致的从机没有接收到主机读取最后一个数据时发送的非应答信号（参考第 3.5 章节）继续占用 SDA 总线，导致主机无法完成 STOP 信号的发送。

若此时 SDA 为低电平，则主机应该发送 9 个 CLK。从机将会在第 9 个 CLK 时释放 SDA 控制权。

若此时 SDA 为高电平，则主机可通过发送 START 信号使从机释放 SDA 控制权。



## 6 版本修订说明

版本号	时间	修订内容
V1.0.0	2019年11月	初始版本
V1.0.1	2020年7月	修改I2C时钟设置描述
V1.0.2	2020年12月	更改为新格式